Operadores para manipulación de bits

Estos operadores tiene distintas funciones en los procesadores, en particular en sistemas embebidos tienen múltiples aplicaciones, algunas de ellas son:

Apagar y prender bits. Se usa cuando queremos activar o desactivar actuadores conectados a los puertos de un sistema embebido. Ejemplo, un relevador que prende o apaga un foco.

Hacer operaciones lógicas con máscaras para manipular datos. Ejemplo, configuración de LCD's en modos de 4 y 8 bits, uso de datos con ADC y manejo de UART's, etc.

Complementar bits. Se usa para generar relojes por software usando una terminal de salida de un puerto digital. Algunas aplicaciones de estos relojes es la generación de señales PWM para manipular servo motores, configuración de frecuencias de corte en filtros programables, generación de sonidos en altavoces, etc.

Determinar el valor de un bit dentro de un registro. Algunas aplicaciones es preguntar por el valor de entrada en una terminal al tener conectado un dipswitch, push button, teclados, etc. Determinar el estado de algún sensor digital como los de efecto hall, fotoresistencias, ópticos, etc. Determinar el estado de algún periférico.

Programar diferentes algoritmos que requieren manipulación de bits como:
1. Algoritmos de detección de paridad.
2. Algoritmos CRC.
3. Algoritmos de cifrado.
4. Algoritmos aritméticos como promediado, multiplicaciones y divisiones.

Los operadores de manipulación de bits se muestran en la tabla 1.

Operación	Operador	Función
Corrimiento a la derecha	>>	Divisiones rápidas por potencias de 2^n
Corrimiento a la izquierda	<<	Multiplicaciones rápidas por potencias de 2^n
AND	&	Apagar bits, se forma una máscara con los bits en 0's que se quieren apagar y 1's en lo bits que no queremos modificar. Determinar el valor de un bit, se forma la máscara con uno en la posición del bit que queremos determinar su valor y 0's en los demás bits.
OR	|	Prender bits, se forma una máscara con los bits en 1's que se quieren prender y 0's en lo bits que no queremos modificar.
XOR	^	Complementar bits, se forma una máscara con los bits en 1's que se quieren complementar y 0's en lo bits que no queremos modificar.
NOT	~	Negar bits, no se usa máscara.

Autor: Victor Hugo García Ortega

Apagar bits. Suponga el número 118, apague los bits 4 y 2.

	7	6	5	4	3	2	1	0
Número = 118 = 0x76	0	1	1	1	0	1	1	0
Máscara = 0xEB	1	1	1	0	1	0	1	1
AND	0	1	1	0	0	0	1	0

```
int main()
{
	char num = 118, res;

	res = num & 0xEB;
	printf(“Numero: %x, resultado: %x”, num, res);
}
```

Prender bits. Suponga el número 118, prenda los bits 3 y 0.

	7	6	5	4	3	2	1	0
Número = 118 = 0x76	0	1	1	1	0	1	1	0
Máscara = 0x09	0	0	0	0	1	0	0	1
AND = 0x7F	0	1	1	1	1	1	1	1

```
int main()
{
	char num = 118, res;

	res = num | 0x09;
	printf(“Numero: %x, resultado: %x”, num, res);
}
```

Complementar bits. Suponga el número 118, complemente el bit 0.

	7	6	5	4	3	2	1	0
Número = 118 = 0x76	0	1	1	1	0	1	1	0
Máscara = 0x01	0	0	0	0	0	0	0	1
XOR = 0x77	0	1	1	1	0	1	1	1

```
int main()
{
	char num = 118, res;

	res = num ^ 0x01;
	printf(“Numero: %x, resultado: %x”, num, res);
	num = res;
	res = num ^ 0x01;
	printf(“Numero: %x, resultado: %x”, num, res);
}
```

Autor: Victor Hugo García Ortega

Determinar el valor de un bit. Suponga el número 118, determine el valor del bit 2.

	7	6	5	4	3	2	1	0
Número = 118 = 0x76	0	1	1	1	0	1	1	0
Máscara = 0x04	0	0	0	0	0	1	0	0
AND = 0x04	0	0	0	0	0	1	0	0

```
int main()
{
        char num = 118, res;

        if( (num & 0x04) == 0x04 )
                printf("El bit 2 esta encendido");
        else
                printf("El bit 2 esta pagado");
}
```

Realizar operaciones matemáticas rápidas. Multiplique un número por la constante 10 usando corrimientos y sumas.

Producto = 10(x)
Producto = x(8+2)
Producto = 8x + 2x
Producto = $x(2^3) + x(2^1)$
Producto = x << 3 + x << 1

```
#define EVER 1

int main()
{
        char num = 11, producto;

        producto = num  << 3 + num  << 1;
        printf("Numero: %d, producto: %d", num, producto);
}
```

Autor: Victor Hugo García Ortega